AN14915

Anti-Rollback Feature on i.MX RT700

Rev. 1.0 — 29 January 2026 Application note

Document information

Information	Content
Keywords	RT700, anti-rollback, security, signed, non-secure image version, secure image version, lock, Secure Provisioning Tool
Abstract	This application note describes the steps for enabling the anti-rollback feature on i.MX RT700 using the Secure Provisioning Tool (SEC tool).

1 Introduction

The i.MX RT700 family integrates anti-rollback protection as part of its secure boot and update mechanisms. This feature ensures that only firmware images with a version equal to or newer than the one previously authorized can be executed. It prevents downgrade attacks such as attempts to load older firmware versions that may contain malware or other known vulnerabilities. By enforcing version checks against values stored in dedicated fuses, the device blocks execution of outdated firmware, as a result, maintaining system integrity and trust.

This protection is part of a broader security architecture in i.MX RT700 devices. The platform also includes (not an exclusive list):

- Secure boot using Elliptic Curve Digital Signature Algorithm (ECDSA) authentication (NIST P-256 or P-384 curves)
- Encrypted firmware updates
- Secure debug based on policies defined by life cycle states
- Secure storage using One-Time Programmable (OTP) or Static RAM-Physically Unclonable Function (SRAM-PUF)
- PRINCE-based memory encryption/decryption (IPED)
- Immutable Root of Trust (RoT) embedded in boot ROM, which enforces life cycle policies and secure provisioning

Together, these features, along with others not listed above, provide a robust foundation for secure embedded applications across development, deployment, and field update phases.

This application note describes the steps for enabling the anti-rollback feature using the Secure Provisioning Tool (SEC tool).

2 Anti-rollback and image version

Anti-rollback feature can be implemented to restrict the usage of an 'older' version of firmware that can have a key compromised or an identified security bug. This in turn avoids the device from the risk of malicious activities. This feature can be used for secure boot and for secure firmware updates. This feature uses the image version number and compares that to the one to be updated. Boot ROM supports this feature for both signed and SB3.1 images.

The i.MX RT700 devices implement anti-rollback protection using dedicated OTP fuse words:

- Secure firmware version (*SEC_FW_VER*): Stored in OTP words 128–131, supports up to 64 version levels (0 to 63). This is typically used for major or critical security updates.
- Non-secure firmware version (*NS_FW_VER*): Stored in OTP words 112–127, supports up to 256 version levels (0 to 255). This is typically used for minor updates.

Both fuses use a bit-counting scheme. For example, 1b for version 1, 11b for version 2, 111b for version 3, and so on. Only the lower 16 bits of these fuse words are active.

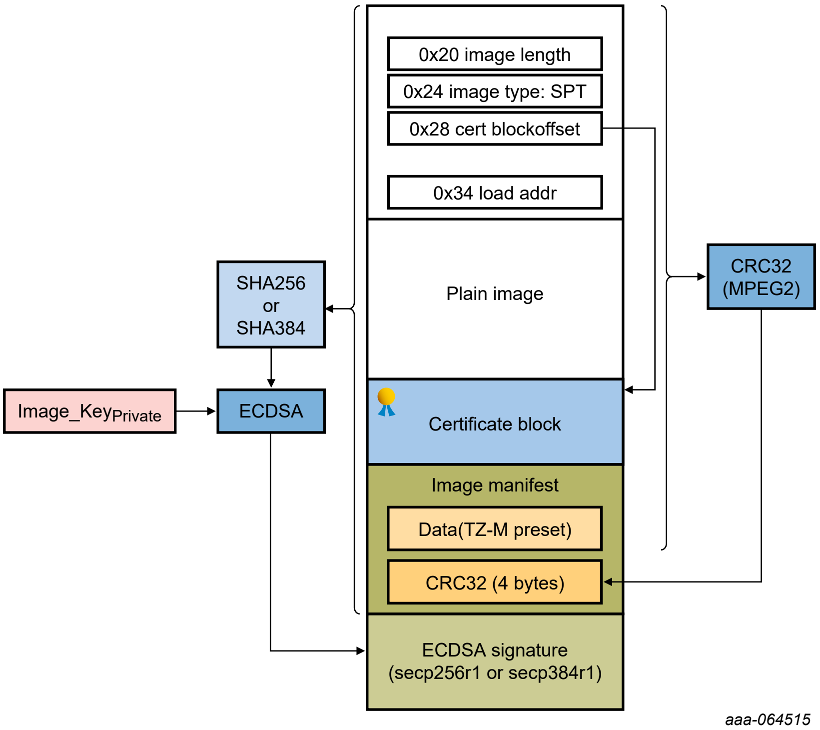

Figure 1. Structure of signed image

In the signed firmware images, the *firmwareVersion* is located in the *Image manifest* block of the signed image (Figure 2). The image manifest begins with a fixed 4-byte string *imgm* (hex value *0x6D676D69*). This string is the *magic* at offset 0x0, to mark the start of the image manifest. The *firmwareVersion* field is found at offset 0x8 and is a uint32_t value in the little-endian format.

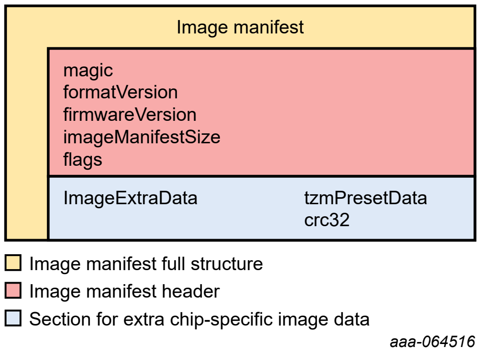

Figure 2. Image manifest

For an SB3.1 image, version checking can be enabled using the `checkFwVersion` command. This command includes a parameter that specifies whether to check the secure version, the non-secure version, or both. The boot ROM first enforces the *SEC_FW_VER* OTP check. It compares the image version stored in the SB3.1 manifest against the value in *SEC_FW_VER*. If the manifest's secure version is lower than the OTP value, the image is rejected immediately, ensuring rollback protection before any SB3.1 command is processed. The `checkFwVersion` command executes only if the image passes the OTP check. This command compares the SB3.1 image version against the specified values (*NS_FW_VER*, *SEC_FW_VER*, or both) as part of the SB3.1 sequence. The SEC tool provides an easy way to embed both secure and non-secure version values into the SB manifest and SB3.1 commands (explained later). This procedure is followed during firmware updates to implement the anti-rollback feature.

For signed images, the ROM compares the image version with OTP *SEC_FW_VER* every time before the application image is booted. The version stored in the *SEC_FW_VER* OTP is compared with the *firmwareVersion* value (present in the signed image). If the *firmwareVersion* value is lower than the value in *SEC_FW_VER*, the image is rejected, enforcing rollback protection. This version check is done along with image validation.

Since the non-secure firmware version typically represents minor updates, it can support up to 256 increments before a secure version change. For instance, an image could start at 1.1 and progress through 1.2, 1.3, and so on until 1.255—representing 256 updates. When a secure update occurs, the version can advance to 2.1. Here, the first digit denotes the secure version and the second digit denotes the non-secure version. This approach can be adopted as a naming convention for firmware updates.

This mechanism ensures that once a device has been updated to a newer firmware version, it cannot be downgraded to an older, potentially vulnerable version. This approach preserves the integrity and security of the system.

3 Locking firmware version

The firmware version fuses can be permanently locked using the fuse bits *SEC_FW_VER_LOCK[2:0]* and *NS_FW_VER_LOCK[2:0]*, which are part of the *LOCK_CFG3* fuse word. This is useful in scenarios where future firmware updates are not required. Once these lock bits are burned, the corresponding version fuses become immutable. Fuse burning is irreversible, so this action should be taken with caution. Table 1 displays the functions based on different values of the *SEC_FW_VER_LOCK/NS_FW_VER_LOCK* bits.

Table 1. Various options for SEC_FW_VER_LOCK/ NS_FW_VER_LOCK[2:0] bits

Value of bits (binary)	Description
000	Unlocked (fuses can be read, sensed, burned, or overridden in the corresponding OTP shadow register)
001	Write Protect (WP). The controlled field cannot be burned.
010	Override Protect (OP). The controlled field shadow registers cannot be overwritten.
011	Override + Write Protect (OP + WP). The controlled field cannot be overridden nor burned.
100	Read Protect (RP). The controlled field can be sensed only, but cannot be read from shadow registers.
101	Read + Write Protect (RP+WP). The controlled field can be sensed or overridden, but cannot be read nor burned.
110	Read + Override Protect (RP+OP). The controlled field can be sensed, burned, but cannot be read nor overwritten.
111	All locks. The controlled field cannot be read, sensed, burned, or overridden in the corresponding OTP shadow register.

4 Demo

Below sections provide information on how to set up the device and steps to demonstrate the anti-rollback feature.

5 Environment

This section gives information on the hardware and software environment.

5.1 Hardware

- Board:
  - MIMXRT700-EVK
- Debugger:
  - Integrated CMSIS-DAP debugger on the board
- Miscellaneous:
  - 1 micro-USB cable
  - PC
- Board setup:
  - Connect the micro-USB cable to the PC through the J54 debug probe.

5.2 Software

- Toolchain:
  - MCUXpresso for Visual Studio Code
- Software package:
  - SEC Tool v25.09 or later

5.3 Demo steps

The below section provides the steps to run the anti-rollback demo.

5.3.1 Prerequisite

Prepare the SEC tool workspace for the i.MX RT700 device. This application note assumes that the user knows how to use the SEC tool and generate a workspace. For more information, refer to *Secure Boot on i.MX RT700* (document AN14821). Figure 3 shows the generated workspace.

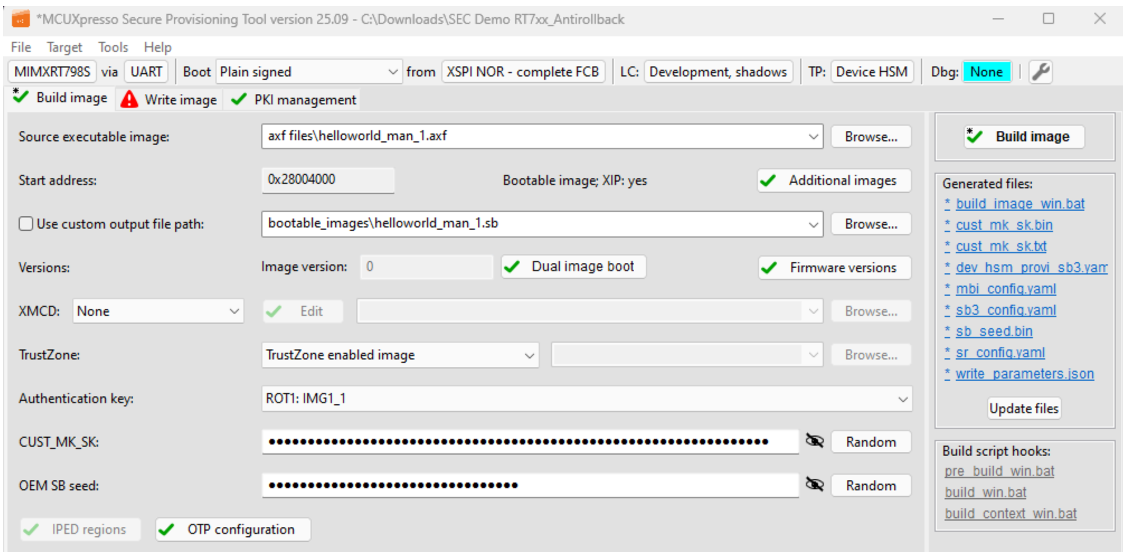

Figure 3. SEC tool workspace for RT700

For reference, the SEC tool workspace used for this demo is available with this application note (AN14915SW.zip). This zip file contains two different application executable files (*.axf) that are generated using the MCUXpresso for Visual Studio Code. These files are based on the *mimxrt700evk_hello_world_cm33_core0* example project that can be downloaded from the SDK. The only difference between the two executable files is in the *printf* statements. The *printf* statements help distinguish between different files by providing the *firmwareVersion* and the *SEC_FW_VER* values. To demonstrate the anti-rollback feature, use these files and follow the steps. These files are provided in the SEC tool workspace under the folder *axf files*.

Also, make sure that the board is in ISP mode using the boot pin mode settings. This can be done using SW10. Table 2 displays the boot mode pin settings using SW10.

Table 2. Boot pin mode settings

BOOT_ISP[1:0]	Boot Type
00 (SW10-x00)	SDHC0 eMMC
01 (SW10-x01)	XSPI0
10 (SW10-x10)	Auto ISP (UART, SPI, I2C, USB)
11 (SW10-x11)	XSPI1

Once SW10 is modified to select Auto ISP option, press SW2 button (SYS_RST) to reset the board and force the device into ISP mode.

5.3.2 Steps for demo

To continue the demo, perform the following steps:

1. Now that the workspace is generated, select the source executable image as **helloworld_man_1.axf** from the folder *axf files* in the attached workspace. Then, click the **Firmware versions** button and a pop up shows up.

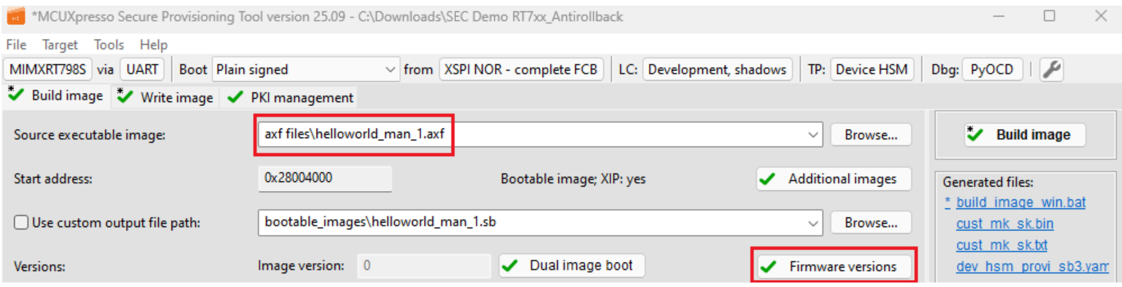

Figure 4. Select the source executable image and the firmware versions

2. In this pop-up, select the **Set minimal firmware version** option to define the minimum secure firmware version required for the device to boot successfully. Set both values – **Image firmware version** and **Set minimal firmware version** to 1, then click **OK**. The value in Set minimal firmware version is copied to the *SEC_FW_VER* OTP fuse. The value in **Image firmware version** is copied to the *firmwareVersion* field in the image manifest.
   Also, once a value is provided for **Image firmware version**, this value is automatically copied to the `checkFwVersion` command in the SB3.1 image.

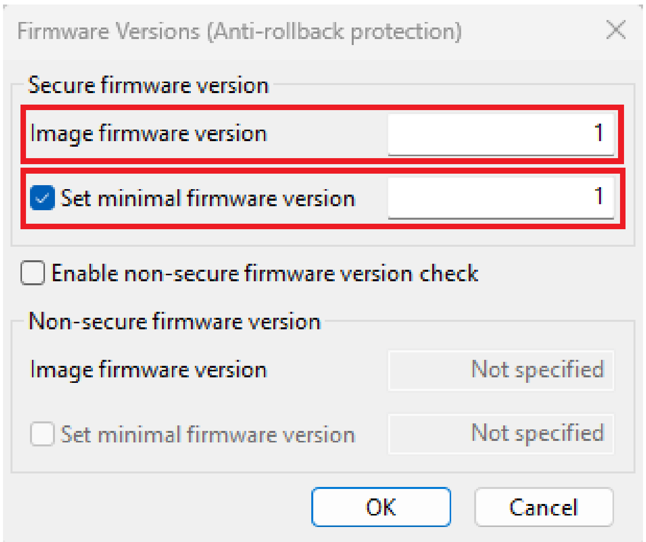

Figure 5. Enable and set the secure firmware version values

Note: *In this demo, we use the secure firmware version only. Non-secure firmware version can also be used. It depends on the usage in the user application. A secure firmware version is typically used for major or critical security updates, while a non-secure firmware version is suitable for minor updates.*

Here, the value of *SEC_FW_VER* OTP is copied to the corresponding shadow register as the life cycle is set to **Development, shadows**.

3. Now, build the image by selecting the **Build image** option. A 'success' message gets displayed after successful operation, as shown in Figure 6.

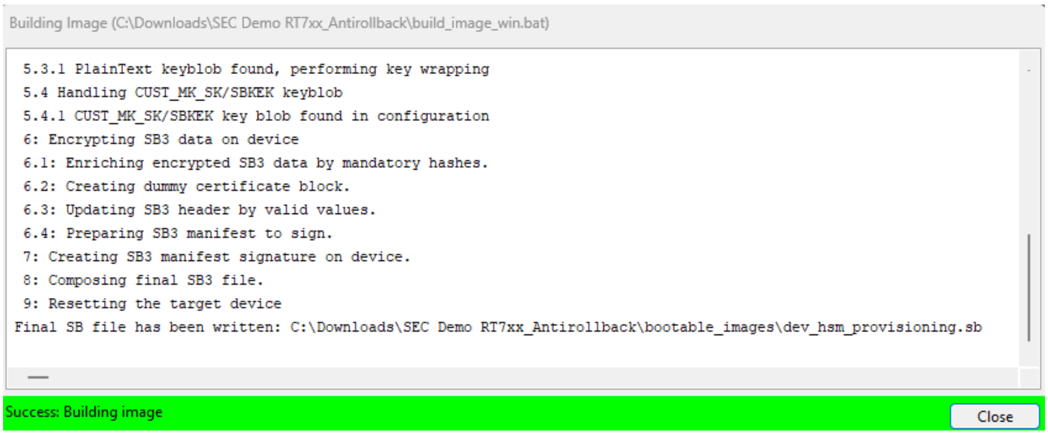

Figure 6. Successfully build image

4. To flash the image onto the device, go to the **Write image** tab and click the **Write image** option on the right side. A 'success' message gets displayed after a successful flashing as shown in Figure 7.

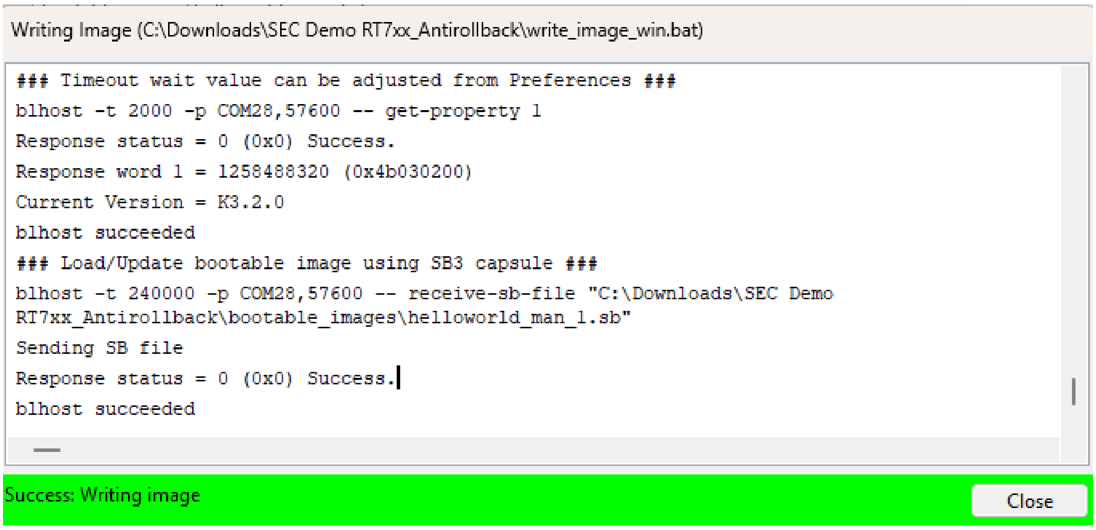

Figure 7. Successful write image operation

5. Now, verify the output displayed in the terminal window. Change the SW10 to XSPI0 mode and press the SW2 button to reset the MIMXRT700-EVK. Open the terminal window and connect the COM port. The below message is displayed in the terminal window. At this point, both values, *firmwareVersion* and the *SEC_FW_VER* in the device are set to "1".

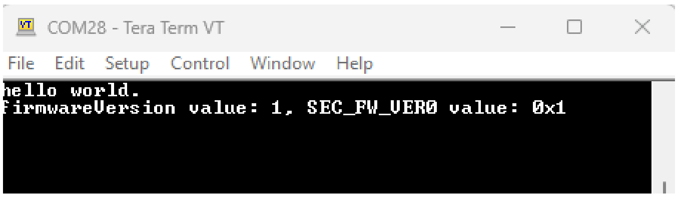

Figure 8. Output from first image

6. Now, prepare another image but this time, set the *firmwareVersion* value as 0 and then try to flash the image on the device. The anti-rollback feature of the device checks whether this image is updated or disregarded. To do that, put the board in ISP mode again using SW10 and press the reset button. Do not give a power cycle as that resets the shadow register values back to default. Also, make sure to disconnect the terminal.
7. Go back to the **Build** tab on the SEC tool and browse for the source executable image. Select the executable image as **helloworld_man_0.axf** from the folder *axf files* in the attached workspace.

Figure 9. Source executable image option

8. Now, go back to the **Build image** tab. Select the **Firmware Versions** option and set both the **Image firmware version** and **Set minimal firmware version** as 0.

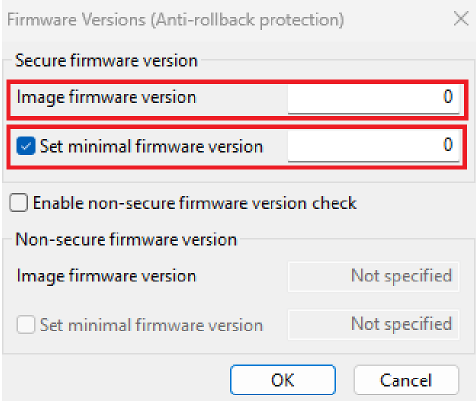

Figure 10. Set Image firmware version as 0

9. Build the image. The success message shows the location of the generated SB file, as highlighted in [Figure 11](#).

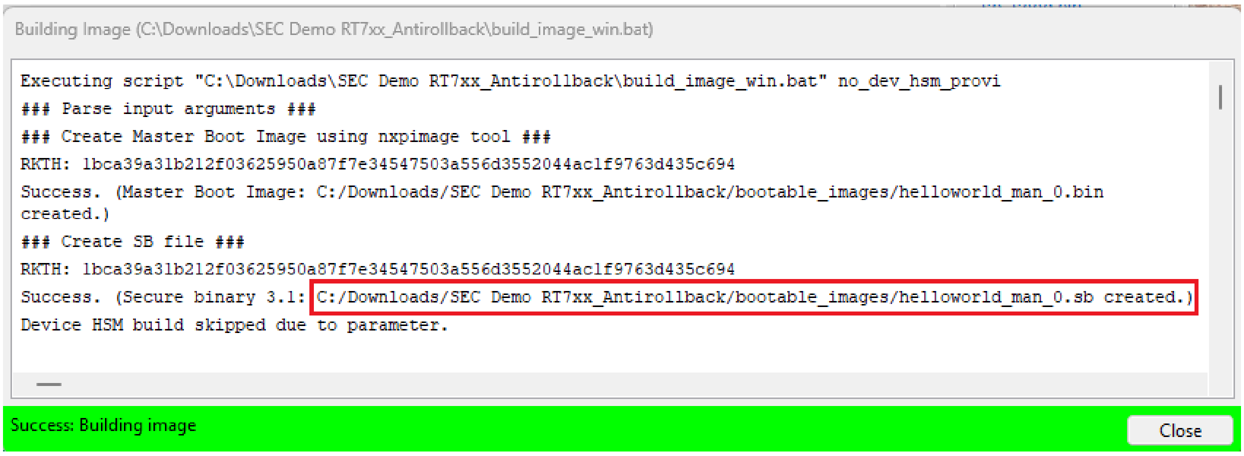

Figure 11. Successfully build image

Now, to mimic a firmware update process, use the blhost tool to flash this generated image on the device. But steps 10 and 11 do not show any debug authentication commands because the device is in the development life cycle state. Debug authentication becomes necessary when the device is in the *Develop2* or a higher life cycle state for a firmware update.

10. Open the command prompt and go to the location of your blhost tool. Type in the `get-property` command to verify the device is in ISP mode and the blhost can talk to the device.

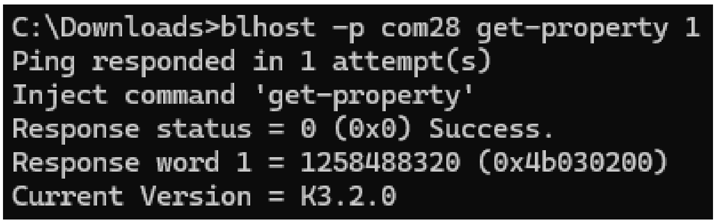

Figure 12. Example blhost command

11. Once a success message is received, type in the below command to update the firmware:

```
blhost -p comXX receive-sb-file <location of generated SB3 file>
```

Where,

XX is the COM port

[Figure 13](#) shows the output after running this command.

```
C:\Downloads>blhost -p com28 receive-sb-file "C:\Downloads\SEC Demo RT7xx_Antirollback\
bootable_images\helloworld_man_0.sb"
Ping responded in 1 attempt(s)
Inject command 'receive-sb-file'
Preparing to send 33792 (0x8400) bytes to the target.
Successful generic response to command 'receive-sb-file'
Data phase write aborted by status 0x2712 kStatus_AbortDataPhase
Possible JUMP or RESET command received.
Response status = 1 (0x1) Failure.
Wrote 0 of 33792 bytes.
```

Figure 13. Failed to update the image

Figure 13 shows that the new image is rejected. It happens during the boot process. The firmware version stored in the SB3.1 header was compared against the *SEC_FW_VER* OTP value. Since the header's secure firmware version is lower than the *SEC_FW_VER* OTP value programmed in the device, the anti-rollback feature prevents the update. As the image was rejected at this initial validation stage, none of the SB3.1 commands, including the `checkFwVersion` command were executed. The device continues to run the original firmware image.

To verify, switch back to XSPI0 mode using SW10 and press the reset button on the board. Open the terminal window and connect the COM port. The output must be the same, as shown in Figure 8, indicating that the update failed.

5.3.3 Scenario with NS_FW_VER

Sometimes, the *NS_FW_VER* OTP might also need to be updated. For such scenarios, the SEC tool provides a seamless process. The non-secure firmware version can be updated using similar options, as described in Section 5.3.2 for secure firmware versions. When non-secure firmware versions are enabled and set, the value is written to the *NS_FW_VER* OTP. The same value is also included in the `checkFwVersion` command in the SB3.1 image.

During the image update process, if the secure firmware version in the SB3.1 image is not lower than the OTP value, the first validation step is passed. After this validation, the SB3.1 commands can run. The non-secure value is checked through the `checkFwVersion` command.

If the non-secure version value in the SB3.1 CheckFwVersion command is lower than the value stored in the device's OTP, the image update is rejected. In this case, a different error is generated. An example of this error is shown in Figure 14. The response status for the error is *kStatusRomLdrRollbackBlocked*.

```
C:\Downloads>blhost -p com28 receive-sb-file "C:\Downloads\SEC Demo RT7xx_Antirollba
ck\bootable_images\helloworld_man_0.sb"
Ping responded in 1 attempt(s)
Inject command 'receive-sb-file'
Preparing to send 33360 (0x8250) bytes to the target.
Successful generic response to command 'receive-sb-file'
(1/1) 1%Data phase write aborted by status 0x2712 kStatus_AbortDataPhase
Possible JUMP or RESET command received.
Response status = 10115 (0x2783) kStatusRomLdrRollbackBlocked
Wrote 512 of 33360 bytes.
```

Figure 14. Error received for non-secure firmware image version

The above demo demonstrates the anti-rollback feature.

6 Acronyms

Section 6 lists all acronyms used in this document.

Acronym	Expansion
SEC	Secure Provisioning Tool
RoT	Root of Trust
ECDSA	Elliptic Curve Digital Signature Algorithm
OTP	One-Time Programmable (Fuses)
SRAM-PUF	Static RAM-Physically Unclonable Function
PRINCE	PRINCE Lightweight Block Cipher
CMSIS-DAP	Cortex Microcontroller Software Interface Standard – Debug Access Port
ISP	In-System Programming
XSPI	Expanded Serial Peripheral Interface (NXP Memory Interface)
COM	Communication Port
SB3.1	Secure Binary Format v3.1 (NXP boot image format)

7 Revision history

Table 3 summarizes revisions to this document.

Table 3. Revision history

Document ID	Release date	Description
AN14915 v.1.0	29 January 2026	Initial public release

Legal information

Definitions

Draft — A draft status on a document indicates that the content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included in a draft version of a document and shall have no liability for the consequences of use of such information.

Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at https://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Suitability for use in non-automotive qualified products — Unless this document expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

HTML publications — An HTML version, if available, of this document is provided as a courtesy. Definitive information is contained in the applicable document in PDF format. If there is a discrepancy between the HTML document and the PDF document, the PDF document has priority.

Translations — A non-English (translated) version of a document, including the legal information in that document, is for reference only. The English version shall prevail in case of any discrepancy between the translated and English versions.

Security — Customer understands that all NXP products may be subject to unidentified vulnerabilities or may support established security standards or specifications with known limitations. Customer is responsible for the design and operation of its applications and products throughout their lifecycles to reduce the effect of these vulnerabilities on customer's applications and products. Customer's responsibility also extends to other open and/or proprietary technologies supported by NXP products for use in customer's applications. NXP accepts no liability for any vulnerability. Customer should regularly check security updates from NXP and follow up appropriately.

Customer shall select products with security features that best meet rules, regulations, and standards of the intended application and make the ultimate design decisions regarding its products and is solely responsible for compliance with all legal, regulatory, and security related requirements concerning its products, regardless of any information or support that may be provided by NXP.

NXP has a Product Security Incident Response Team (PSIRT) (reachable at PSIRT@nxp.com) that manages the investigation, reporting, and solution release to security vulnerabilities of NXP products.

NXP B.V. — NXP B.V. is not an operating company and it does not distribute or sell products.

Trademarks

Notice: All referenced brands, product names, service names, and trademarks are the property of their respective owners.

NXP — wordmark and logo are trademarks of NXP B.V.

Contents

Date of release: 29 January 2026
Document identifier: AN14915